Kapitel 1 Domæner

Et domæne er et en samling af værdier med operatorerne plus og gange. Eksempelvis er R et domæne, ligesom N er et domæne. Dette kapitel vil dække domæner og deres egenskaber.

I kapitel 1 findes en tabel med diverse domæner. Disse domæner kan sidestilles med OOP, hvor en klasse kan indeholde nogle bestemte properties.

Domæne	Indhold
$\mathbb{R}$	Alle reelle tal $\{\ldots, -2, -1.816, 0, 1, \sqrt{2}, \pi, \ldots\}$
$\mathbb{C}$	Komplekse tal $\{i, j, z\}$
$\mathbb{N}$	Naturlige tal $\{1, 2, 3, \ldots\}$
$\mathbb{Z}$	Hele tal $\{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$
$GF(2)$	Et domæne der kun indeholder 0 og 1

Tabel 1.1: Tabel over nogle af de eksisterende domæner.

1.1 Komplekse tal

Komplekse tal, er tal der ikke nødvendigvis kan forklares. Disse tal indeholder blandt andet tal som i, j og z, og beskrives med domænet $\mathbb{C}$. Tallet i bruges blandt andet til at forklare ligningen

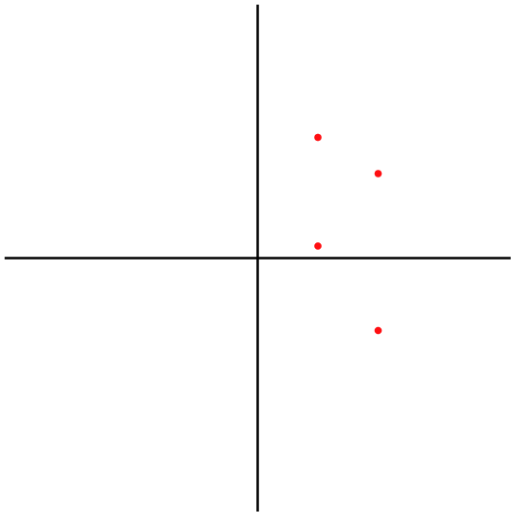
$$x^2 = -1 \tag{1.1}$$

Med ligningen i Ligning (1.1) kan $16i^2 = -16$ nu udregnes. Med dette kan det konkluderes at $4i$ er løsningen til $x^2 = -16$, da $(4i)^2 = -16$. Ligeledes kan ligningen $(x-1)^2 = -16$, løses ved $x - 1 = 4i$, hvilket giver at $x = 1 + 4i$. Derfor kan det konkluderes at alle komplekse tal har en reel del og et imaginær del.

1.1.1 Værdier i komplekse tal

Ligesom almindelige punkter, kan plottes i en graf, kan komplekse tal også plottes. Betragt følgende eksempel (Figur 1.1) fra listing 1.1:

Kodeuddrag 1.1: Python kode til plotte komplekse tal i en graf.
Koden kan findes i ./labs/fields/complex_plot.py



Figur 1.1: Et plot af komplekse tal, i en liste S.

```
1  from plotting import plot
2
3  S = [1+0.2j,1+2j,2+1.4j,2-1.2j] # Make set
4
5  [print(abs(S[i])) for i in range(len(S))]
6  # The length from the origin of each element in the set are
7  # 1.019803902718557
8  # 2.23606797749979
9  # 2.4413111231467406
10 # 2.33238075793812
11
12 plot(S, 4) # plot image
```

Punkterne kan plottes da komplekse tal har en reel værdi samt en imaginær værdi. Af samme grund kan et komplekst tal også have en længde fra origo, hvilket fremkommer af linjerne 6–10 i listing 1.1. Dette grundes at på y aksen måles måles punktets imaginære værdi, mens den reele værdi måles på x aksen. Det sidste er simpel pythagoras.

1.2 Abstraktion af domæner

Abstraktion kan være en nødvændighed i matematik, og ikke mindst programmering. Eksempelvis kan ligningen $ax + b = c$, hvor a ikke er nul. Dette kan gøres programmatisk på følgende vis:

Kodeuddrag 1.2: Python kode til at udregne $ax + b = c$, hvor a ikke er nul.
Koden kan findes i ./labs/fields/abstraction.py

```
1  def solve(a,b,c): # Solve for x in ax+b=c
```

```
2       return (c-b)/a
3
4
5   print(solve(10,5,6)) # 0.1
6   print(solve(-3,2,9)) # -2.333->
7   print(solve(4+2j,7,3)) # (-0.8+0.4j)
8   print(solve(0,23,68)) # Error as a = 0
```

Som det også fremkommer af listing 1.2 kan komplekse tal også bruges i python. Disse komplekse tal skal dog noteres som j.

Kapitel 2 Sandsynlighedsteori

2.1 Sandsynlighed

Sandsynligheden, når man kaster en terning, kan beskrives med følgende formel:

$$S = \{H, T\} \tag{2.1}$$

Dette kaldes udfaldsrummet, og der den værdi, der beskriver sandsynligheden, for $\frac{1}{n}$. Det er en endelig mængde, og kan omskrives til

$$S = \{a_1, a_2, \ldots, a_n\} \tag{2.2}$$

Når man vil finde udfaldet af k elementer, i en liste med længden n, hvor man trækker tilfældige elementer, kan mange bruge binomialfordelinger. Når man regner binominialkoeficienter, findes der flere forskellige måder at udregne disse på.

x	Uden gentagelser	Med gentagelser
Uden orden	$\binom{n}{r} = \frac{n!}{r!(n-r)!}$	$\binom{n+r-1}{r} = \frac{(n+r-1)!}{r!(n-1)!}$
Med orden	$\frac{n!}{(n-r)!}$	r^n

Her er n mængden, og k er antallet af elementer, som vi vil finde.

2.1.1 Kortspil

I et normalt deck af kort, findes der 52 kort, uden jokeren. Her kan man regne på, hvad sandsynligheden for at trække et bestemt kort er, ved at udregne permutationer, med ingen gentagelser. I følgende eksempel trækkes 4 kort.

$$\binom{52}{4} = C(52, 4) = \frac{52!}{4!(52-4)!} = 270,725 \tag{2.3}$$

Dette vil altså sige at der er en 277,725 mulige udfald, af de fire kort. For at finde sandsynligheden for at trække 4 Esser, kan man simpelt skrive $\frac{4}{52}$.

2.1.2 Lotto!

I et spil lotto udtrækkes der r numre af n tal. Hvis vi vil finde sandsynligheden for at det første tal er rigtigt, men resten er forkerte, kan vi bruge følgende formel:

$$\frac{\binom{r}{1} \cdot \binom{n-r}{r-1}}{\binom{n}{r}} \tag{2.4}$$

I et tilfælde hvor vi har med en mængde på 48 at gøre, og der hver gang trækkes 6 tal, kan vi skrive formlen som følgende:

$$\frac{\binom{6}{1} \cdot \binom{42}{5}}{\binom{48}{6}} = \frac{212667}{511313} = 0.41592 \tag{2.5}$$

Monty Hall problemet

I et gameshow bedes en deltager om at vælge en af tre døre. Bag en af disse døre er der en præmie. Når deltageren har valgt en dør, åbner værten en dør, hvor præmien ikke er bag ved. Monty Hall problemet siger at hvis deltageren skifter dør, efter at værten har valgt, vil resultere i $\frac{2}{3}$ chance for at vinde præmien. Dette kan bevises matematiks ved følgende udtryk

$$P(A) = \frac{1}{3} \tag{2.6}$$

$$P(C|B\ tom) = P(A^c) = \frac{2}{3} \tag{2.7}$$

2.1.3 Terningekast

Hvis man kaster en terning, hvor siderne 2 og 4 er vægtede og dermed giver en 3 gange så høj chance som at kaste med de andre sider, kan formlen opskrives som følgende:

$$\begin{aligned}
P(2) + P(4) &= 3(P(1) + P(3) + P(5) + P(6)) \\
P(2) &= P(4) \\
2P(2) &= 12P(1) \\
P(2) &= 6P(1) \\
P(1) + P(2) + P(3) + P(4) + P(5) + P(6) &= 1 \\
16P(1) &= 1 \\
P(6) = P(5) = P(3) = P(1) &= \frac{1}{16}
\end{aligned}$$

2.2 Bayes Sætning

Hvis vi antager at vi ved hvad antallet af events er, kan vi med Bayes sætning, bestemme sandsynligheden, for et af samme events. Dette kan sidestilles med emails.

Vi ved hvor mange emails der er spam. Med dette kan vi bestemme sandsynligheden for at næste email er spam.

Sætning 1 (Bayes Sætning)**.**
Antag at E og F er events fra vores sample S, således at $p(E) \neq 0$ og $p(F) \neq 0$. Så kan vi sige

$$p(E|F) = \frac{p(E|F)p(F)}{p(E|F)p(F) + p(E|\overline{F})p(\overline{F})} \tag{2.8}$$

Dette kan også omskrives til

$$p(E|F) = \frac{E \cap F}{p(F)} \tag{2.9}$$

Beviset for denne sætning skal skrives som følgende

Bayes Baseform opskrives som

$$p(F|E) = \frac{p(E|F)p(F)}{p(E)} \tag{2.10}$$

I et tilfælde hvor vores $p(E) = \frac{1}{3}$, $p(F) = \frac{1}{2}$ og $p(E|F) = \frac{2}{5}$. Her vil vi finde $p(F|E)$.

$$\frac{\frac{2}{5} \cdot \frac{1}{2}}{\frac{1}{3}} = \frac{3}{5} = 0.6 \tag{2.11}$$

2.2.1 Betinget sandsynlighed

Kapitel 3 Vektorer

Dette kapitel vil beskrive todimensionale vektorer som 2-vektorer, og tredimensionelle vektorer som 3-vektorer. Der vil blive brugt bracket notation for vektorer. Det er her fordelagtigt at introducere Det Euklidiske Rum.

Definition 1 (Det Euklidiske Rum)**.**
Det Euklidiske Rum, nogle gange kaldet det Kartesianske rum, eller **n**-rummet, er rummet, der indeholder alle n-tupler, af reelle tal ($\{x_1, x_2, \ldots, x_n\}$). $\mathbb{R}^n$ er derfor et vektorrum, og betegnes også som n-vektorer. Derved har man at $\mathbb{R}^1 = \mathbb{R}$, $\mathbb{R}^2$ er Den Euklidiske Plan.

Det fremkommer af Det Euklidiske Rum, at en 4-vektor, kan skrives som $\mathbb{R}^4$. Dette er da Definition 1 siger, at n-vektorer, skal være $\mathbb{R}^n$.

Definition 2 (Vektorer)**.**
En vektor er i geometrien et objekt, der er defineret ved at have en længde og en retning.

3.1 Geometrien i sæt af vektorer

Dette afsnit har til formål at beskrive det geometriske aspekt af vektorer. Det vil dække spænding af vektorer.

3.1.1 Spændinger af vektorer over $\mathbb{R}$

$$\text{Span}\{v\} = \{\alpha v \mid \alpha \in \mathbb{R}\} \tag{3.1}$$

Ovenstående er et sæt, der kun indeholder lineære kombinationer, af en ikke-nul-vektor v. Hvis et tomt sæt af 2-vektorer skal beskrives, findes en vektor i sættet, en nul-vektor. Denne vektor er beskrevet som $[0, 0]$, i vektor notation.

Samtidig beskriver det ovenstående sæt også, at der for en 2-vektor, findes uendelig mange punkter i eksempelvis $\{\alpha[2, 3] \mid \alpha \in \mathbb{R}\}$, hvor vektoren kan

I en spænding af $\{[1, 0], [0, 1]\}$, er de to vektorer standard for $\mathbb{R}^2$, hvorved hver 2-vektor er i spændingen. Dette betyder at sættet indeholder alle punkter i Den Euklidiske Plan.

En 4-vektor kan også betegnes som et sæt af elementer eller en funktion. Betragt følgende vektor $\{2.87, 6.2, 3.7, 1.5\}$. Dette kan betegnes som funktionen

$$0 \mapsto 2.87 \tag{3.2}$$
$$1 \mapsto 6.2 \tag{3.3}$$
$$2 \mapsto 3.7 \tag{3.4}$$
$$3 \mapsto 1.5 \tag{3.5}$$

Samme 4-vektor kan repræsenteres som en Python Dictionary $\{0 : 2.87, 1 : 6.2, 2 : 3.7, 3 : 1.5\}$.

3.1.2 Sparsitet

En vektor hvis værdier alle er 0, kaldes en sparsom eller nul-vektor. Hvis ikke mindre end k værdier i vektoren er 0, kaldes vektoren en k-sparsom vektor. Vektorer der repræsenterer data, der er opsamet af sensorer, er næste aldrig sparsomme vektorer.

3.2 Hvad kan vises med vektorer

Mange ting kan vises med vektorer. Blandt disse ting er *Binære strenge*, altså strænge at 1 og 0, der har en given værdi. Disse vises ved vektorer som $\{1, 0, 0, 1, 1, 0, 1\}$, hvor vektoresn længde, er det samme som længden af den binære streng. *Attributer* kan også vises med vektorer. Dette er som vist i Ligning (3.2), hvor to værdier kan sidestilles som en Python Dictionary. Mere relevant for sandsynlighedsteori, kan vi også repræsenterer *Sandsynligheds distribution* med vekoterer. Dette vises igen som i Ligning (3.2), hvor hver key er mapped til en værdi. *Billeder* kan også vises som vektorer, da hvor farve har en RGB værdi, som kan afkodes til en binær værdi. Ydermere kan punkter i rummet plottes som vektorer, dog er dette givet ved Mat A adgangskursus, så dette behøver ikke yderligere forklaring.

3.3 Vektor addition

Vektor addition foregår som på Mat A adgangskurset. Dog vil denne sektion gå mere i dybten med vektor addition.

Definition 3 (KLEIN s. 69)**.**
Addition af n-vekoterer er definieret ved addition af de respektive værdier.
$[u1, u2, u3] + [v1, v2, v3] = [u1 + v1, u2 + v2, u3 + v3]$

Siden vektorer som $[1, 2]$ kan bruges som et opslag, kan den tænkes som at overføre en værdi. Eksempelvis fra $[4, 4]$ til $[5, 6]$. Derfor må den også kunne adderes.

Samtidig findes der regler for addition af vektorer, eksempelvis at

$$x + y = y + x \tag{3.6}$$

Samt

$$(x + y) + z = x + (y + z) \tag{3.7}$$

3.4 Skalarproduktet

Ligesom med addition er dette også gennemgået på Mat A adgangskurset. Multiplikation af en vektor, med en skalar α, svarer til at gange på hver værdi i vektoren. Dette kan eksemplificeres med vektoren $[3, 4, 7]$. Hvis denne ganges med skalaren 5, gøres dette på følgende vis:

$$5[3, 4, 7] = [5 \cdot 3, 5 \cdot 4, 5 \cdot 7] = [15, 20, 35] \tag{3.8}$$

Hvis der samtidig adderes en vektor på, følger dette de arritmetiske regler. Der ganges først, så adderes der. Hvis en vektor ganges med et negativt tal, ændres fortegnet for alle elementer i vektoren. Dette betyder at vektoren vil ændre orientering. Samtidig gælder følgende for multiplikation af vektorer med ekstra skalarer:

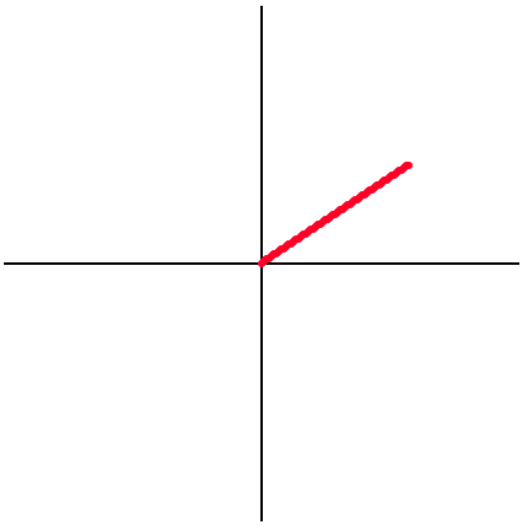
$$\alpha(\beta v) = (\alpha\beta)v \tag{3.9}$$

3.4.1 Linjer gennem origo

Med vektorer kan linjer også tegnes fra origo. Dette kan blandt andet gøres med følgende sæt af punkter.

$$\{\alpha v \mid \alpha \in \mathrm{R}, 0 \leq \alpha \leq 1\} \tag{3.10}$$

Ovenstående sæt tenger en linje af punkter fra origo til v. Med python koden vist i listing 3.1 gives følgende plot

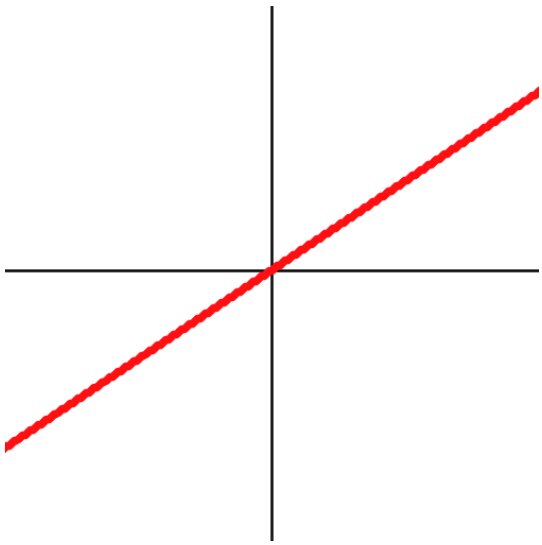


Figur 3.1: Plot fra udsnittet listing 3.1

Kodeuddrag 3.1: Python kode til at generere plot i Figur 3.1.
Koden kan findes i ./labs/vectors/python_plot.py

```
4 def scalar_vec_mult(a, v): # Function to multiply vector
5     return [a*v[i] for i in range(len(v))]
6
7 plot.plot([scalar_vec_mult(i/100, [3,2]) for i in range(101)], 5) #
      plot image
```

I kombination med at segmenter af linjer, kan samles til hele linjer og at der kan dannes uendelig mange skalar vektorer, kan vi danne figurer, når α strækker sig over flere tal. Skalarprodukter < 1 giver anledning til kopier af v, samtidig med at negative Skalarprodukter giver anledning til vektorer der peger i den anden retning. Sættet af punkter $\{\alpha v \mid \alpha \in \mathrm{R}\}$ strækker sig fra $-\infty; \infty$, og producerer derfor følgende plot:



Figur 3.2: Sættet $\{\alpha v \mid \alpha \in \mathrm{R}\}$ plottet i en graf. Både x og y aksen går fra $-\infty; \infty$

Grafen i Figur 3.2 kan laves med følgende stykke python kode:

Kodeuddrag 3.2: Python kode til at generere plot i Figur 3.2.
Koden kan findes i ./labs/vectors/scalar_plot.py

```
4 def scalar_vec_mult(a, v): # Function to multiply vector
5     return [a*v[i] for i in range(len(v))]
6
7 plot.plot([scalar_vec_mult(i/100, [3,2]) for i in range(-301,301)],
      5) # plot image
```

3.5 Kombination af addition og skalar

Med information fra afsnit 3.3, 3.4 og 3.4.1 kan det dedukteres at man med vektorer kan plotte alverdens figurer. Dette betyder selvfølgelig også at man kan begynde at lege med mere avancerede former for vektore beregning. Betragt nedenstående

Kapitel 4 Matricer

En matrice over $\mathbb{F}$ er en to-dimensionel liste af elementer af typen $\mathbb{F}$. En matrice vises på følgende måde:

$$\begin{bmatrix} 1 & 2 & 3 \\ 10 & 20 & 30 \end{bmatrix}$$

Denne matrice består af to rækker og tre kolonner, der alle indeholder en værdi. Dette kaldes en 2×3 matrice, og indeholder rækkerne [1 2 3] samt [10 20 30]

Hvis en traditionel matrice A skal repræsenteres med rækker og kolonner, kan dette gøres med en kolonne-liste L. Dette skrives åp følgende måde:

$$A\,[i,j] = L\,[j]\,[i] \text{ for hver } 0 \leq i < m \text{ og } 0 \leq j < n. \tag{4.1}$$

Igen kan matricen $\begin{bmatrix} 1 & 2 & 3 \\ 10 & 20 & 30 \end{bmatrix}$ bruges. Med ovenstående procedure skrives matricen som $[[1, 10], [2, 20], [3, 30]]$.

4.1 Vektorer som matricer

For at definere en matrice ud fra en *D-vektor* over $\mathbb{F}$, gives det at $R \times C$ er en matrice over $\mathbb{F}$. Her er det kartetiske produkt $R \times C$. Denne matrice kan repræsenteres som:

	@	#	?
a	1	2	3
b	10	20	30

Figur 4.1: "Eksempel på en vektor matrice."

I afsnit 4.1 er rækkernes labels givet langs siden, og kolonnernes labels er listet ovenfor matricen. Ydermere kan matricen, med python kode, beskrives på følgende måde

Kodeuddrag 4.1: Python kode udsnit

```
1     >>> {('a','@'):1, ('a','#'):2, ('a','?'):3, ('b','@'):10, ('b',
          '#'):20, ('b','?'):30}
```

4.2 Rækker, kolonner og elementer

Betragt listing 4.1 og afsnit 4.1. Disse matricer kan begge konstrueres i python på

Kapitel 5 Definitioner og Beviser

5.1 Definitioner

Definition 4 (S. 440, 1)**.**
Antag at S er et sæt med n elementer. Den uniforme fordeling, tildeler sandsynligheden $\frac{1}{n}$ for hvert element i S

Definition 5 (S. 440, 2)**.**
Sandsynligheden for en hændelse E er summen af sandsynligheder, af udfaldet af E. Dette er

$$p(E) = \sum_{s \in E} p(s) \tag{5.1}$$

Bemærk at når E er et uendeligt sæt, så er $\sum_{s \in E} P(s)$ en konvergent uendelig serie.

Definition 6 (SLIAL02, S. 9)**.**
Hvis vi har givet et sandsynlighedsfelt med udfaldsrum S. Da siges X at være en stokastisk variabel (random variable) hvis X er en funktion med definitionsmængde S og med værdier i R.
Eksempel: Terningkast. $S = \{1, 2, 3, 4, 5, 6\}$. $p(i) = \frac{1}{6}$

Lad $X(i) = i$, for alle $i \in S$. Eksempel: n uafhængige Bernoulli forsøg.
Lad $X(s) =$ antal succeser, hvor s er en række af n Bernoulli forsøg.
Middelværdien af denne kan udregnes ved følgende formel

$$E(X) = \sum_{s \in S} p(s)X(s) \tag{5.2}$$

Definition 7 (S. 442, 3).
Lad e og F være hændelser med $p(F) > 0$. Den konditionale sandsynlighed af E, givet ved F, noteret med $p(E|F)$, er defineret som

$$p(E|F) = \frac{p(E \cup F)}{p(F)} \tag{5.3}$$

Definition 8 (S. 443, 4).
Hændelserne E og F er uafhængige, hvis og kun hvis $p(E \cap F) = p(E)p(F)$

Definition 9 (S. 444, 5).
Hændelserne $E_1, E_2, \ldots$ er parvis uafhængige, hvis og kun hvis $p(E_i \cap E_j) = p(E_i)p(E_j)$ for alle par af heltal i og j, hvor $1 \leq i \leq j \leq n$.
Disse hændelser er gensidig uafhængige, hvis

$$p(E_{i_1} \cap E_{i_2} \cap \cdots \cap E_{i_m}) = p(E_{i_1})p(E_{i_2}) \cdots p(E_{i_m}) \tag{5.4}$$

når $i_j, j =, 1, 2, \ldots, m$ er heltal, hvor $1 \leq i_1 < i_2 < \cdots < i_m \leq n$ og $m \geq 2$.

5.2 Beviser og sætninger

Sætning 2 (S. 441, 1).
Hvis $E_1, E_2, \ldots$ er en sekvens af parvis disjunkte hændelser i et sample rum S, så kan vi sige:

$$p\left(\bigcup_i E_i\right) = \sum_i p(E_i) \tag{5.5}$$

Bemærk at denne sætning anvendes når sekvensen $E_1, E_2, \ldots$ består af en finit mængde af tal, eller en tællelig uendelig mængde af tal, der er parvis disjunkte hændelser